EXHIBIT D

RealSecure 1.0

User Guide and Reference Manual

Chapters:

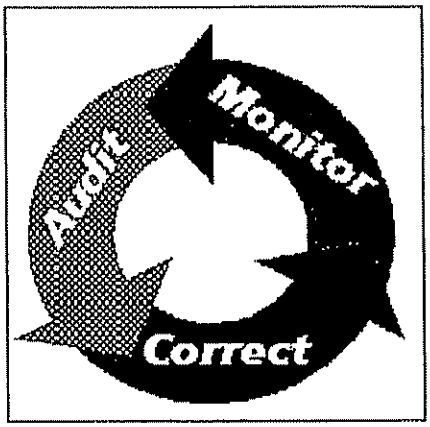
Appendix:

For technical support, email ISS at: rs-support@iss.net

Acknowledgments

Introduction to Real-Time Intrusion Detection

Security is a hot topic today, especially on the Internet. But very few people understand how to achieve an acceptably secure system. Some companies sell products which promise to make a network secure. What they neglect to mention is that there is no single solution to security. Security is a process and a way of doing business which must continually be updated.



Security Cycle

The picture above shows the cycle used to dynamically improve the security of a system. First, an auditing tool is used to find the holes in a network. Combining a security audit with a security policy establishes the original baseline security. Second, the parts of the system which failed the audit are corrected. The system can then be audited again and again to ensure that new holes don't appear or old ones resurface. Third, a monitoring tool is used to watch for new security breaches or attempts to abuse old holes. This keeps the security manager in touch with the state of the network.

RealSecure falls into the third category. It resides on a computer connected to the network and watches the traffic that goes by. This allows it to compare traffic to a wide variety of attack signatures. It summarizes the information in a concise manner so that the security manager can understand what is happening on the network, *as it happens*.

Common Uses for Network Intrusion Detection

RealSecure is very versatile and thus has can perform many different functions. Some possible uses include:

- Auditing network utilization. (How many hits do each of our Web servers get? Who is transferring files from our FTP site?)
- Auditing and blocking network intrusion attempts. (How many attacks do we get from non-US sites? How many `rlogin` attempts do we get and from where?)
- Providing a second tier of security behind a firewall. (We should never see `telnet` sessions coming into our internal network. So kill and log any such attempts.)
- Detecting network state and possible problems. (Did Sales just put a new machine on our

network? Who is using it? Did they accidentally use an IP address of an existing machine?)

- Obtaining profiles of a detected intruder and assessing damage. (OK, we now know that the intruder is using our Web server to attack other sites. We will log all of his session data throughout the night and play it back later.)

Remember that RealSecure can analyze **any** traffic which passes by. Thus, if you are going to watch for invalid login attempts, RealSecure will print out all invalid login attempts that are made over its network. This is a great improvement over normal auditing in which the administrator has to search a log from each system on the network for suspicious activity.

Dangers

Because RealSecure watches and responds to network events, it has some associated risks. First, it can log all data in a connection including keystrokes and e-mail. Discretion should be exercised when using this feature. Once logs are created, they should be kept on a secure host as they may contain passwords or other sensitive data. This is another reason to run RealSecure on a dedicated machine. Intruders will see the monitoring machine as a prime target, both to steal its saved data and to erase evidence of their actions.

Second, RealSecure responds to events. In particular, the 'kill' option, if misused, can block traffic over an entire network. Be careful to ensure that any connections being killed are the ones that are supposed to be blocked. A wildcard rule with a 'kill' action will block all connections, including `http`, `telnet`, and `ftp`.

The rule of thumb to use is "think twice, configure once".

License

Legality

In most cases, the US government has upheld the right of individuals to monitor their own networks. The general consensus is that all users should be notified of the monitoring. We recommend that you consult a lawyer if you have any questions as to the legality of this product's use. See this CERT advisory explaining the legal issues.

Next section: Installing RealSecure

Installing RealSecure

To obtain maximum benefits from RealSecure, it should be installed on a dedicated machine at the entry point to the network. Good places include the Ethernet interface just inside the firewall or between the Internet router and the internal machines. For security and performance, it is strongly recommended that RealSecure be run on its own dedicated machine. The machine should ideally have as many of its own services as possible turned off. Also, the only users should be the administrator(s). RealSecure collects a lot of data from the network, so the more power and disk space the machine has, the better.

System Requirements

To run the RealSecure engine, you need:

- SunOS 4.1.x, Solaris 2.3 and up, or Linux (kernel versions 1.3.x and up)
- An Ethernet interface connected to the target network
- 486-class performance or better. Note that if the machine is not dedicated to running RealSecure, more resources will be necessary.
- At least 25 MB of free disk space

To use the RealSecure GUI you also need :

- The X-Window system, version 11 or higher
- Motif Installation (Solaris 2.3 and 2.4 only)

How to Install RealSecure

Step 1: Get the Distribution Software

From the ISS Web site.

Go to http://www.iss.net/RealSecure to download a copy for your system.

On CD-ROM

A. Mount the CD-ROM volume (see your Unix manual for instructions).

B. Copy `/mount/point/rs-<OS>.tar` to the install destination directory (where `<OS>` is the name of your operating system). For instance, if the mount point is `/cdrom`, the destination directory is `/usr/local`, and the operating system was SunOS 4.1.3, the command would be as follows:

```
# cp /cdrom/rs-SunOS.tar /usr/local
```

Step 2: Copy the Distribution Software to the Destination System

If you are going to be running the GUI on Host A and engines on hosts A, B, and C, then you need to transfer the `tar` file to each of those systems. Ways to transfer the archive include FTP and e-mail.

Perform the remaining steps on *each* machine running RealSecure.

Step 3: Untar the Archive and Run the Install Program

Change to the directory where you put the `tar` file. For example:

```
# cd /usr/local
```

Then enter these commands:

```
# tar xvf rs-*.tar
# cd rs
# ./install.rs
```

Step 4: Start sssd on Each System Running the RealSecure Engine

In order to run the RealSecure engine, you need to install `sssd` on the host. If you plan to run the RealSecure engine on the same machine as the GUI, you only need to perform these tasks for the machine you just installed RealSecure on. If you want to run the RealSecure engine remotely or you are going to run multiple engines in order to sniff multiple segments or networks, you need to perform these tasks for each machine you are going to run the engine on.

A. Set Up the sssd Engine

Log in to the machine you are going to run the engine on as `root`.

Create a file containing the name or IP address of each machine on your network you are going to run the RealSecure GUI on, along with a random pass phrase that will be used to authenticate the connection.

This authentication file specifies which hosts are allowed to connect to `sssd` to start up engines and a pass phrase used during connections for authentication and encryption. The default location of this file for the `sssd` server is `/etc/sssd.auth`. This can be changed by specifying the `-a` option along with an alternate pathname for the authentication file on the command line when running `sssd`.

The format of the authentication file is:

```
<Pass Phrase>//<hostname>
```

`<hostname>` can be a domain name or IP address. `<Pass Phrase>` should be a unique, hard-to-guess set of letters, numbers, and punctuation. Do **not** use a single word, a sentence or sentence fragment from a well-known published book or song, or anything else that could possibly be guessed. The `sssd` server runs an engine that is capable of watching all the traffic on your network, and access through the `sssd` server is a very serious compromise to the security of a system. Because IP addresses can sometimes be spoofed, it is vital to the security of your system that you choose a good pass phrase. The authentication file must be owned by `root` and **not** be readable by users on the system other than `root`. Make sure you create the file with proper modes **before** entering pass phrases.

Example:

```
# whoami
root
# cd /etc
```

```
# echo ad98IU Aj2 ah c89 kqaknsdh//myguibox.mydomain.com > sssd.auth
# chmod 600 /etc/sssd.auth
```

B. Set Up the GUI

On the machine you will run the RealSecure GUI from, you need to create an authentication file containing the hostnames and pass phrases of each host running `sssd` that you wish to run a RealSecure engine on. This includes the local host if you only plan to run local engines.

The default filename for this file is `/etc/sss.auth`. The entries in this file must match the pass phrase in the `/etc/sssd.auth` file on each machine for which an entry is created. (If you are only running a local engine, you can accomplish this by simply copying your `sssd.auth` file to `sss.auth`.) As with the `sssd.auth` file, you should make sure that nobody other than `root` can read the contents of the file.

Because the machine the RealSecure GUI is run on can control the remote engines on all the machines in the `sss.auth` file, it is extremely important that the machine remains secure against attack. If at all possible, the GUI should be run on a machine with no untrusted users and no services running.

For example:

```
# whoami
root
# cd /etc
# echo ad98IU Aj2 ah c89 kqaknsdh//ssdbox1.mydomain.com > sss.auth
# echo okPui uz 472 JK cnzx opzutb//sssdbox2.mydomain.com >> sss.auth
```

C. Start sssd

To start the `sssd` daemon from a boot startup file, put the following command in a system `rc` file your system uses (i.e.,`/etc/rc`, `/etc/rc.local`, or `/etc/rc.2/SXX.sss`, etc.). The file you will put it in depends on your operating system:

```
sh -c "(cd <RealSecure_distrib_directory>; ./sssd -s )"
```

Where: `<RealSecure_distrib_directory>` is the directory path in which you installed the RealSecure engine.

`sssd` can also be started from a shell command prompt by typing in the same command given above while logged in as `root`. It can be run as a foreground task by using the `-f` option, and if you want verbose output as to what it is doing, use the `-v` option along with `-f`.

Here is a summary of command line options for `sssd`:

`-a <file>`	Authentication file and path (default: `/etc/sssd.auth`)
`-f`	Foreground processing (do not run as daemon)
`-p <port>`	Port number to listen for connections
`-s`	Log connections to `syslog`
`-v`	Verbose output

Stopping sssd

The `sssd` daemon can be stopped with the `kill` command. For example:

```
# kill -TERM `ps ax | grep sssd | grep -v grep | cut -f1 -d" "`
```

Your arguments to `ps` may vary.

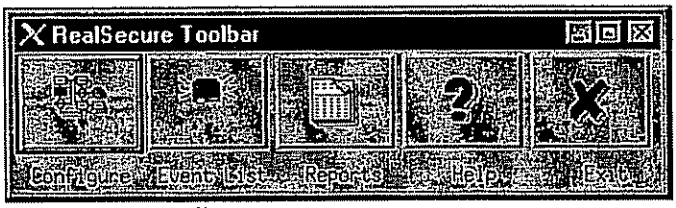
Note: It is necessary to start the `sssd` server even if you are only running an engine on your local machine.

Step 5: Start Real Secure GUI (rsgui) on the One Management Machine

Enter these commands:

```
# cd rs
# rsgui
```

The RealSecure toolbar is displayed.



RealSecure Toolbar

The toolbar allows you to:

- Configure RealSecure engines
- View network events
- Generate reports

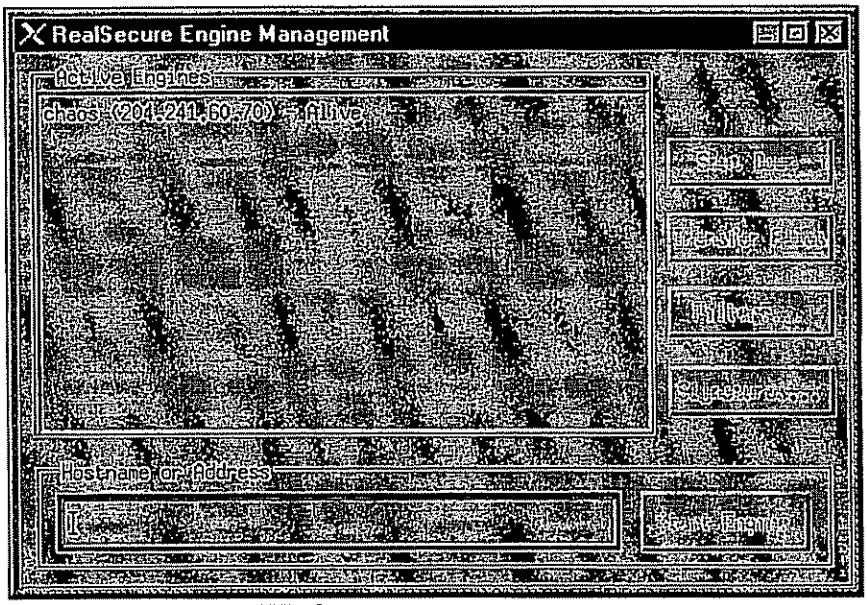
Next section: **Configuring RealSecure**

Configuring RealSecure

To effectively monitor your network for problems, RealSecure needs to know what your network is like. For instance, a Web server should be getting Web traffic, but your company's accounting machine probably shouldn't. Thus, RealSecure would be configured to ignore Web traffic to the Web server, but log any other Web traffic. Each service accessed through your network should be evaluated the same way.

RealSecure has two configuration modes that are related but separate. The filter configuration mode sets what services RealSecure will watch for connections. The feature configuration mode enables, disables, and fine-tunes its custom attack signatures.

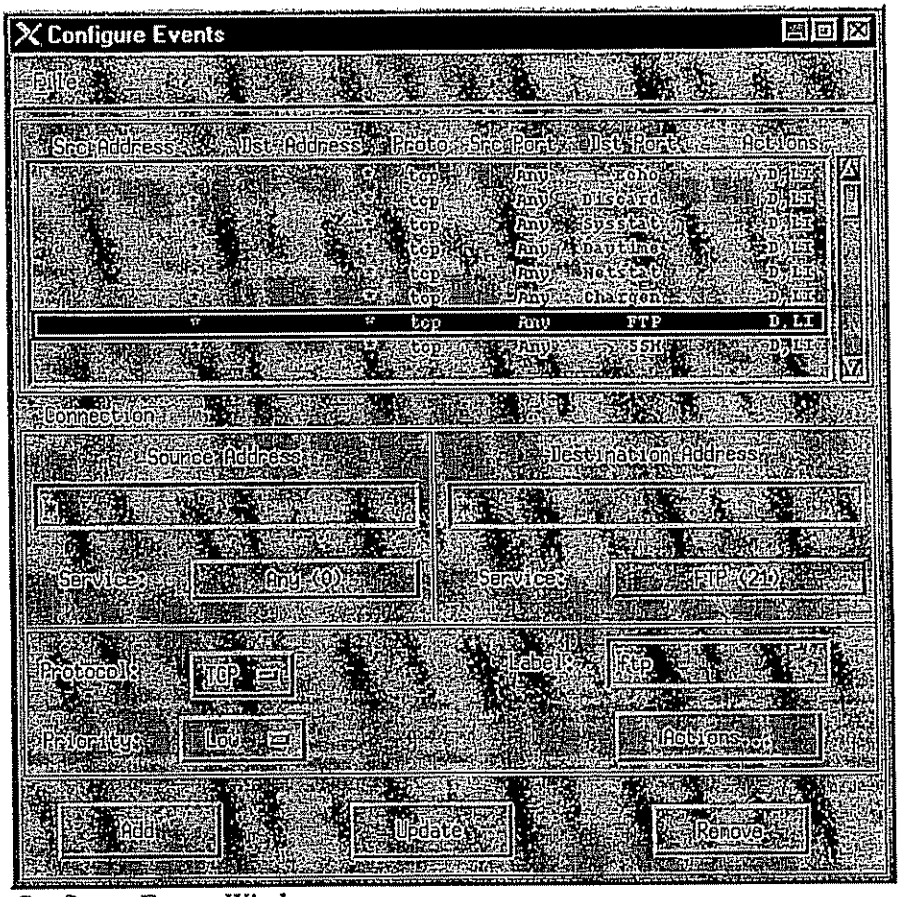
The starting point for GUI configurations is the Engine Management window. To display this window, choose Configure Engines from the RealSecure toolbar.



Engine Management Window

Filter Configuration

The filter configuration does exactly what its name implies--select which types of network connections RealSecure should ignore or watch. To edit the filter configuration, click on the engine you wish to configure in the Engine Management window and then click on **Filters**. The Configure Events window is displayed. **Note:** Alternatively, you can edit the `filter.cfg` file on the engine's host.



Configure Events Window

A filter is made up of rules. The rules in the filter are matched in order. Once a match is found, further searching is cancelled. If the engine you are configuring has a filter configuration, the rules appear in the Filter List . To modify a rule, click on the rule. It will appear in the bottom half of the window. Clicking on any of the fields allows you to modify them. Clicking on the **Modify** button commits your changes to the list. To add a rule, click on the position in the list where you would like to insert your new rule. Enter your desired rule in the fields in the bottom half of the window. Then, click the **Add** button to add the new rule.

There are several fields to a rule:

- Source and Destination Address - IPs or ranges of IPs.
- Source and Destination Service - TCP/UDP ports
- Source and Destination Type - ICMP only
- Protocol - TCP, UDP, or ICMP
- Label - a one-word description of the event that appears in the GUI
- Priority - the severity of this rule
- Actions - what to do when this rule is matched

Addresses

The source and destination addresses are in the common dotted decimal form (i.e. 10.1.2.3). To specify a range of addresses, use the asterisk (*) wildcard. For instance, an address of 10.1.1.* would match

10.1.1.2 and 10.1.1.50, but not 10.1.18.2. Wildcards must be on even boundaries. For example, 10.* is valid, but 10.1.1* isn't. Finally, a wildcard by itself will match all addresses.

Services

Services are the ports in a connection. For instance, HTTP (Web) traffic uses port 80. To select a service, click on the button. A list of services will appear. Select the one you want and click OK. Selecting the **Any (0)** service will match any service. If the service you want is not in the list, edit the `services` file included with the distribution software to add your desired service.

ICMP Types

Every ICMP packet has a type and sub-type. For instance, `ping` packets have an ICMP type of Echo Request. To select a type, click on the button. A list of types will appear. Select the one you want and click OK. If the service you want is not in the list, edit the `services` file included with the distribution software to add your desired ICMP type.

Protocols

Each rule must be of a specific protocol type. Valid protocols are **TCP**, **UDP**, and **ICMP**. TCP is a reliable data transport used for services like E-Mail, FTP, and HTTP. UDP is a datagram service used for services like CU-SeeMe and Talk. Lastly, ICMP is used for sending control messages between Internet nodes.

Labels

The label is a one-word tag for this particular rule. It appears in the logs, as well as on the display. It allows you to differentiate connections at a glance. Valid tags include `Web-Traffic` or `Bob's_PC` but **not** `My Server` (note the space).

Priority

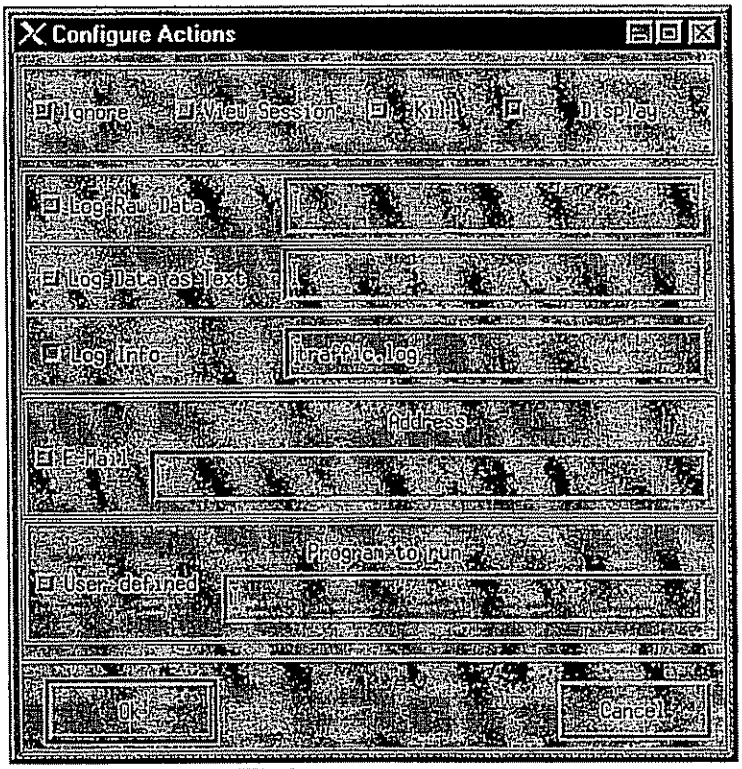
Each rule has a priority, which controls which window the event appears in, as well as grouping events for generating reports. Valid priorities are **high**, **medium**, and **low**. You will want to group rules by priority to weed out common events (Web transfers, E-Mail) from less common events (attempts to exploit security holes, connections to the accounting machine).

Actions

The action configuration is the same for both filters and attack pattern matching. There are quite a few possible actions:

- Ignore any event that matches this rule
- Display a message in the main window that the event occurred
- View the data from the connection in real-time
- Kill the connection by sending a reset packet (only possible with TCP connections)
- Mail a notification to the administrator
- Run a User-specified program when the event occurs
- Log data to a file:
    - Log Info that the connection occured to a file

- Log Text data sent through the connection
- Log Raw data sent through the connection for later playback



Configure Actions Window

To enable or disable an action, click on it. Note that when the **Ignore** action is enabled, all other actions are disabled. Some actions require additional data. For instance, if the **Mail** action is enabled, RealSecure must have an address in order to send mail. Once you have selected the desired action(s), click OK to continue.

Sample Uses of Filter Rules

Getting an Idea of Your Network Traffic

To initially get used to filters and get a feel for what your network traffic is like, use a `filter.cfg` file with the following two entries:

```
tcp     0.0.0.0/0    0.0.0.0/0   0    0   All-TCP   2    D
udp     0.0.0.0/0    0.0.0.0/0   0    0   All-UDP   3    D
```

Then, make sure you are `root`. Follow the startup instructions for using RealSecure. This configuration will display all TCP and UDP connections on your network. The display can get very crowded, quickly. Select **Quit**.

Now, review what your network policy is. Do you allow `rlogin` from anywhere? From just internal

hosts? Examining the firewall's configuration can help here. The entries in the `filter.cfg` file are matched one at a time, in order. A match means the action specified at the end of the line is taken, and further matches are discarded. For this reason, wildcard entries should be saved for the end.

From your security policy, determine what the filter rules will be for your site. Usually, these will correspond with your firewall configuration. Thus, RealSecure can be used as a second level of defense. If your firewall fails, RealSecure will notice and show exactly what unauthorized traffic is occurring.

Click here to see a general configuration that shows all common services.

Since RealSecure engines communicate with the GUI host via normal UDP packets, RealSecure should be configured to ignore those. Here are some rules to ignore all RealSecure reports destined for the GUI host:

```
udp  0.0.0.0/0  10.0.0.1/32  835   835  RSgui   3   ignore
udp  0.0.0.0/0  10.0.0.1/32  836   836  RSeng   3   ignore
```

Sample Configurations

To help you get started using RealSecure, three sample configurations are included. To use them, copy the sample `filters.cfg` and `features.cfg` to each of the hosts that will be using them. Here's a description of each configuration:

- All filter/ features - Turns on all RealSecure checking and data decoding
- Exp filter/ features - Turns on just the exploit checking (leaves out some of the general information features)
- Web filter/ features - Turns on just the Web checks

Click here for detailed description of these sample configurations.

Feature Configuration

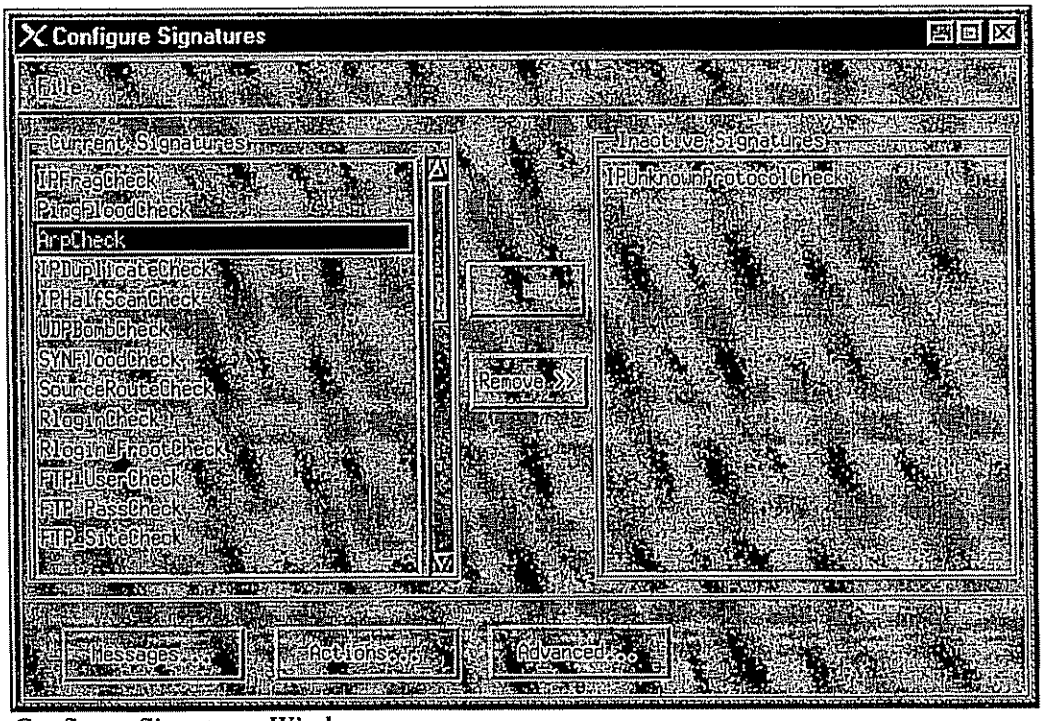
RealSecure has a standard set of attack signature checks that look at the data inside a connection and check for the tell-tale signs of intrusion attempts. For instance, one of the old Sendmail exploits involved passing a malformed **From** address in an e-mail header. RealSecure watches all SMTP connections for this bad data and triggers an alarm if it sees such an attack. Then, the network administrator can investigate and use the logging and response options of RealSecure to track down and lock out the intruder.

To enable this signature checking, the filter configuration must have an entry for the service. For instance, if the above-mentioned Sendmail bug check was turned on, but port 25 (Sendmail) wasn't being watched, the check would never be used.

For a list of the checks and an explanation of what they do, see RealSecure Features and Attack Signatures.

Each check has a standard set of options, configurable through the GUI. Select the engine to configure in the Engine Management window and then click on the **Signatures** button. The Configure Signatures

window is displayed. Alternatively you can edit the `features.cfg` file.



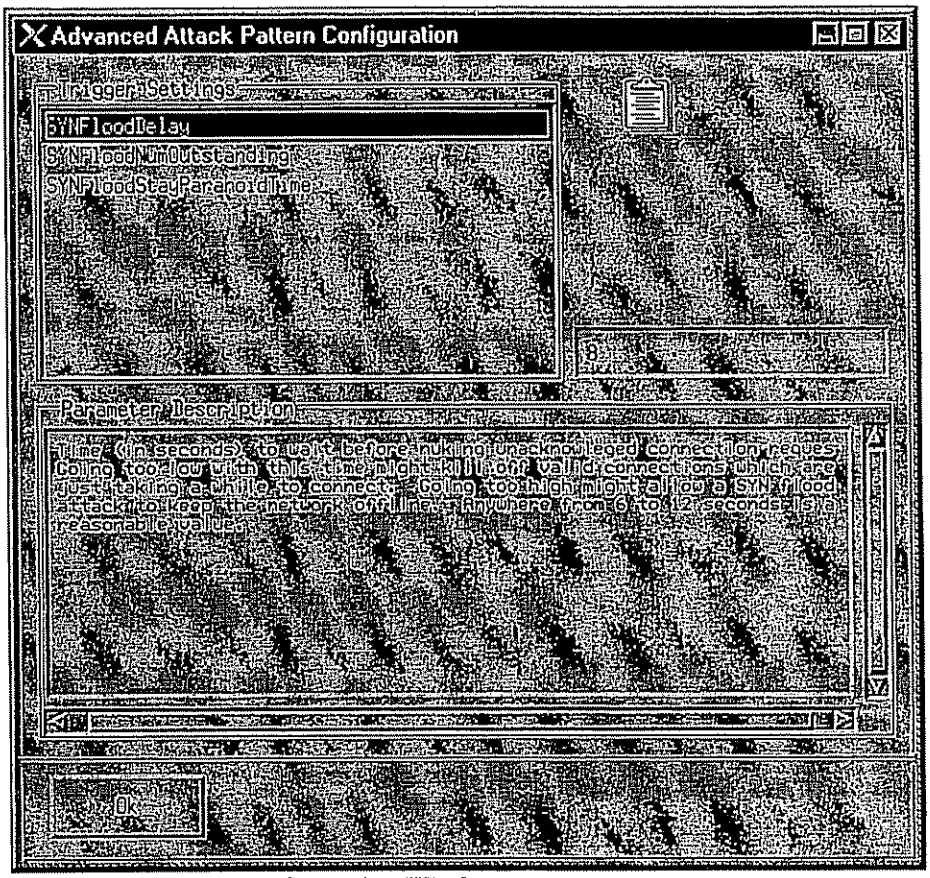
Configure Signatures Window

Each check has a standard formatting for its options. For instance, the check for IP fragmentation attacks has the prefix **IPFrag**. To change the priority of this check, you would modify the entry called **IPFragPriority**.

Here is a list of the options common to all checks:

- ...Check - whether to perform this check (yes or no)
- ...Priority - the priority level for this attack
- ...SMessage - the message to log when this attack happens
- ...Actions - a list of actions to take when RealSecure sees this attack

There are also tunable parameters to some of the checks. These allow you to set the sensitivity of these checks to prevent false alarms. RealSecure's default configuration has reasonable values for these settings, but it is quite possible they will be different for your network. These are configurable by selecting the check in the Configure Signatures window and clicking on **Advanced** button.



Advanced Pattern Configuration Window

Checks

IP Fragmentation

This check looks for IP fragments with a size less than or equal to **IPFragThreshold.** Note that this parameter is in multiples of 8, so to check for an offset less than 16, you would use a setting of **2.** There shouldn't be any need to change the default value.

Ping Flooding

This check looks to see if more than **PingFloodPackets** are received in **PingFloodDelta** seconds. The default setting is 50 packets in 3 seconds. If your network is on a slow connection like 14.4 PPP, you might want to make this setting more sensitive. Otherwise, the default value should suffice.

Arp Check

If a host is down and someone tries to contact it, multiple address request packets will be sent with no response. **ArpMaxUnAcked** sets how many requests are sent to an unresponsive host before triggering an alarm.

Synflood Check

The SYN flood check has two levels of response -- normal and paranoid. **Normal** should be the state during average to heavy network traffic. **Paranoid** mode should only be activated when the system is definitely under a malicious SYN flood attack.

The **SYNFloodDelay** parameter sets how long to wait before resetting SYNs that have not been SYN-ACKed by the server. During normal operations, a SYN should be SYN-ACKed within milliseconds. Under heavy load, it might be possible for a system to take several seconds to respond. **SYNFloodNumOutstanding** is the second parameter needed to trigger paranoid mode. It sets how many SYNs are left waiting on a port before resetting them. Using these two parameters in concert may take a little tuning work. For instance, with a Delay setting of 10 and a NumOutstanding setting of 30, RealSecure will wait until a port (say E-Mail) has 30 connection attempts that have gone unanswered for 10 seconds before going into paranoid mode.

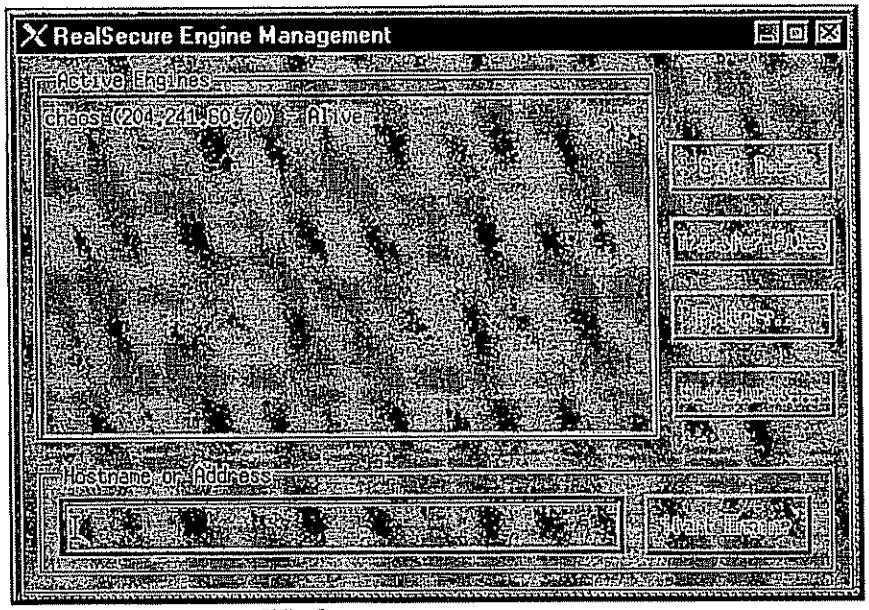
In paranoid mode, RealSecure kills off packets which match the flood packets immediately. Some loss is allowed to permit retransmitted packets (from a valid source) to have a good chance of making it through. However, paranoid mode can significantly affect the performance of your network, so it should not be used unless you are definitely under attack. For this reason, it is recommended that you set RealSecure to be relatively insensitive to SYN floods and install the appropriate vendor patches. These two measures in concert will allow you to know when a SYN flood is occurring yet not be affected by the attack.

SYNFloodStayParanoidTime sets how long to stay in paranoid mode after the SYN flood attack (as defined by the previous two settings) is over. If you have a slow network link, leaving this setting high will clean up after the attack has ended, resetting any lingering connection attempts. If the SYN flood attack restarts, RealSecure goes back into paranoid mode.

Next section: Using RealSecure

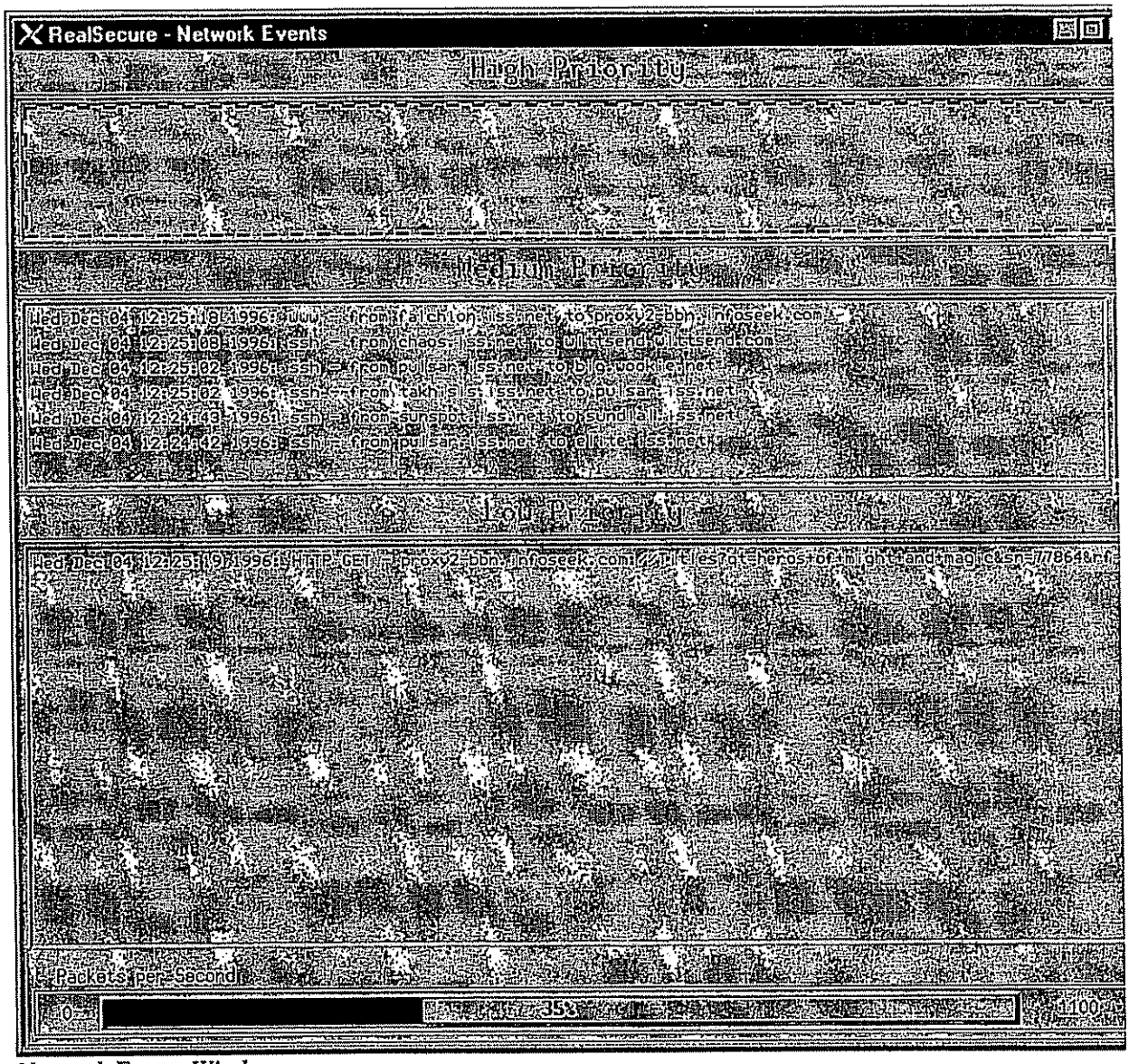
Using RealSecure

When you select **Configure Engines** from the RealSecure toolbar, the Engine Management window appears. This window allows you to start, stop, and configure engines on each host. If there are already engines running on your network, they will automatically contact the GUI and appear in this list within ten seconds.



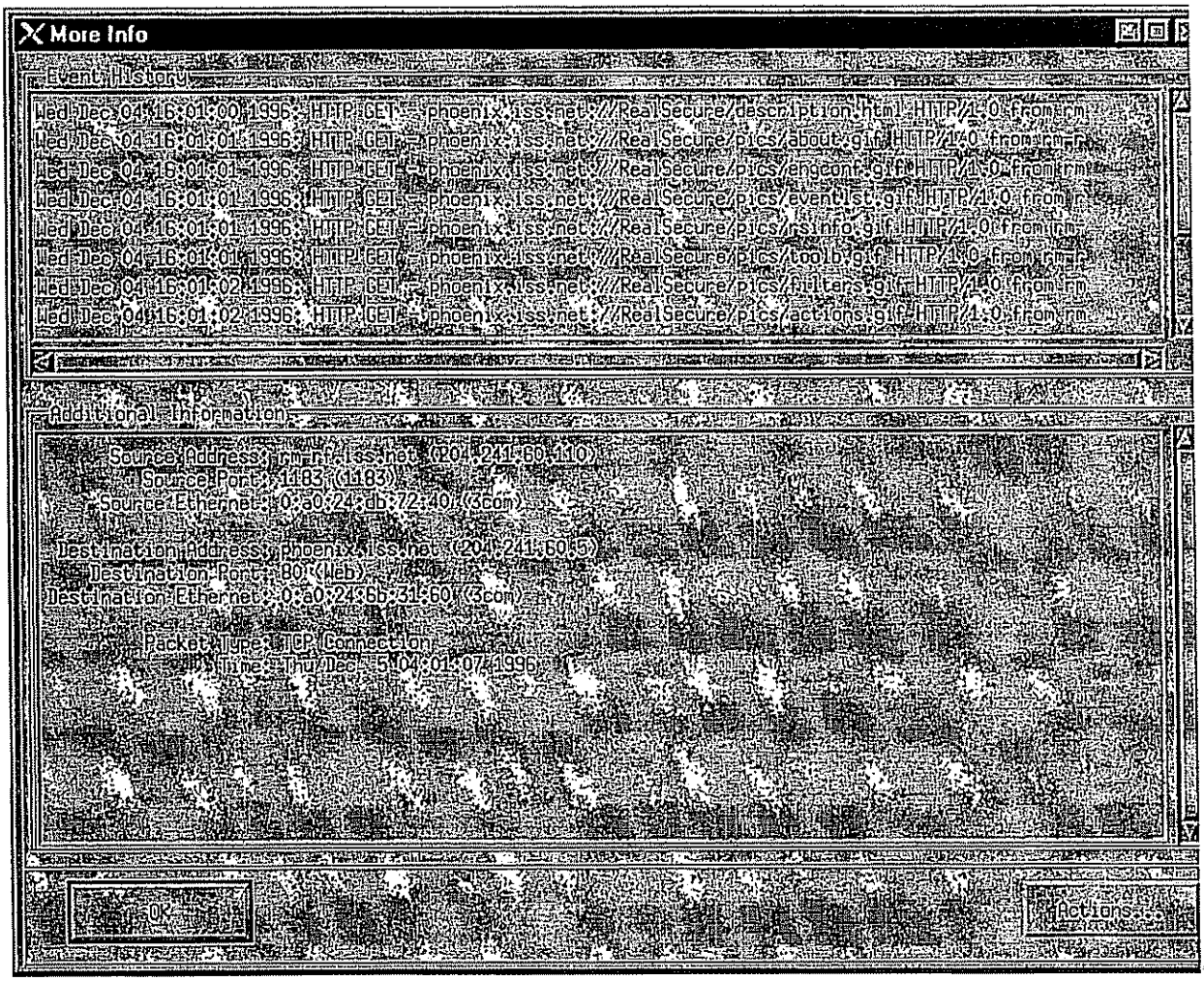
Engine Management Window

To start an engine, type the name or IP address of a host on which you wish to run an engine and press Enter, or click on **Start Engine**. After a short delay, the engine will appear in the engine list and the Network Events window will pop up to start displaying events.



Network Events Window

As events appear in the window, you can get more information about them by double-clicking on the entry.



More Info Window

The More Info window displays information about the event, like the Ethernet addresses and TCP/UDP ports associated with the connection. Also, a list of all related events allows you to track a connection from start to finish. To close this window, click on OK. To perform some action with this connection, click on **Actions**. Note that certain actions are limited by the network state or the underlying protocol. For instance, you can't kill a connection that doesn't exist any more.

The bar at the bottom of the screen displays the number of packets per second seen by the engine(s). If they bar ever goes above 100%, that becomes the new maximum.

Events are added to the top of the list. Old events are moved toward the bottom. Each window (high, medium, and low) has a timeout value for its events. When the time expires for an event, the event is removed from the screen. If it was logged to a file, you still have a record of the event that will show up on any reports you generate. If the event occurs again after being removed, the new event will be displayed.

As you watch events occur, you may find that there are some which you do not wish to see. Go back to the configuration screens and set those events to be ignored. Alternatively, you may find that you need to see some events you have been ignoring. RealSecure is meant to change with your changing network

needs. So feel free to modify or use the sample configurations. Make sure to keep a backup copy of the known good configurations.

Next section: Generating Reports

Generating Reports

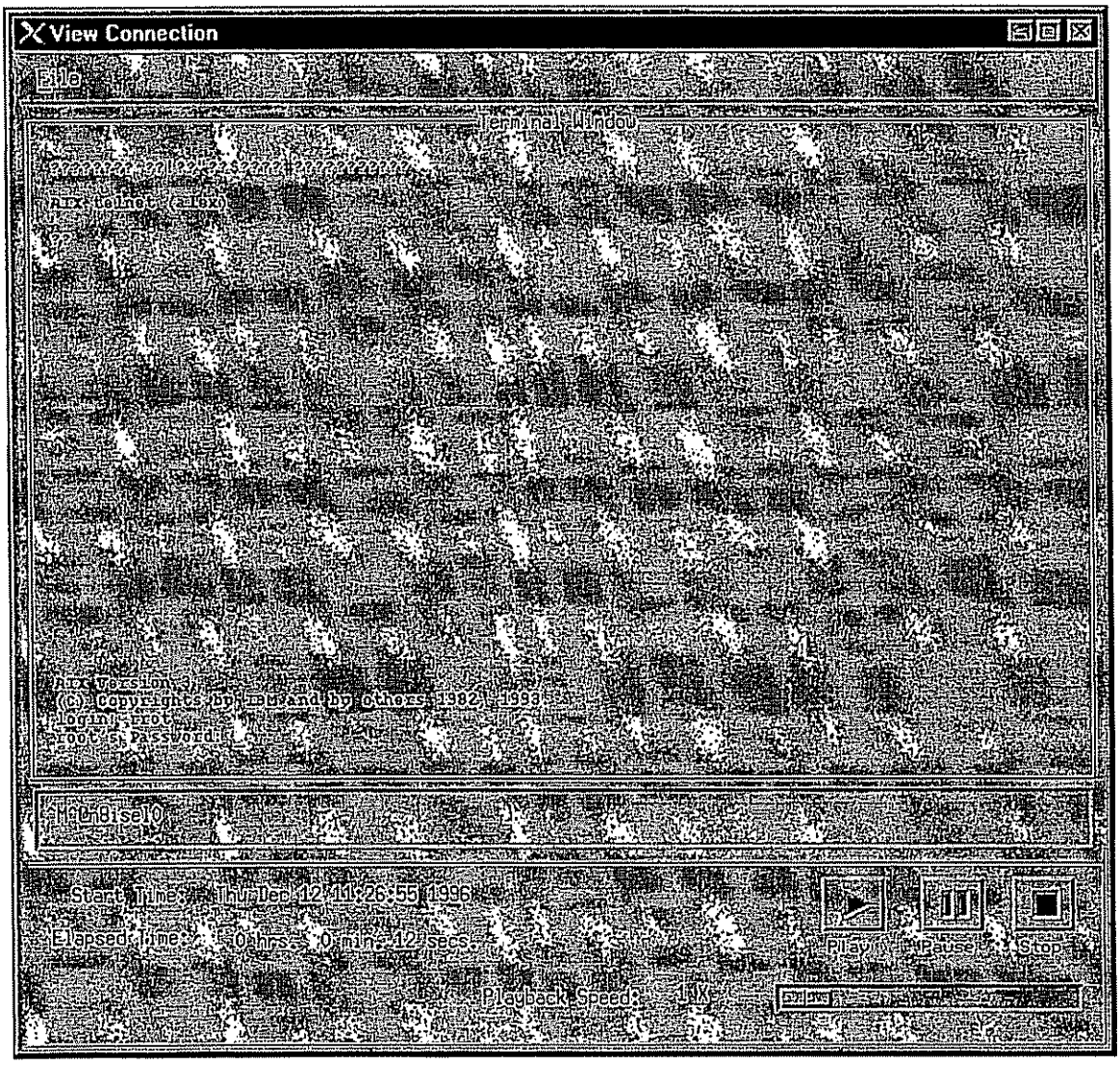
After generating various logs of events, you will probably want to see a summary of network attacks. This can help you see where your major network problems are and modify security configurations appropriately. It is also useful for seeing trends in attacks and preparing a strategy for handling future intrusions.

Playback Feature

Although not a reporting feature *per se*, the **playback** feature of RealSecure allows it to show you what the intruder saw and typed. To use this feature, get a log generated with the **Log Raw** action and enter:

```
# playback <my-logged-raw-data-file>
```

You will see a list of the connections recorded in that file.

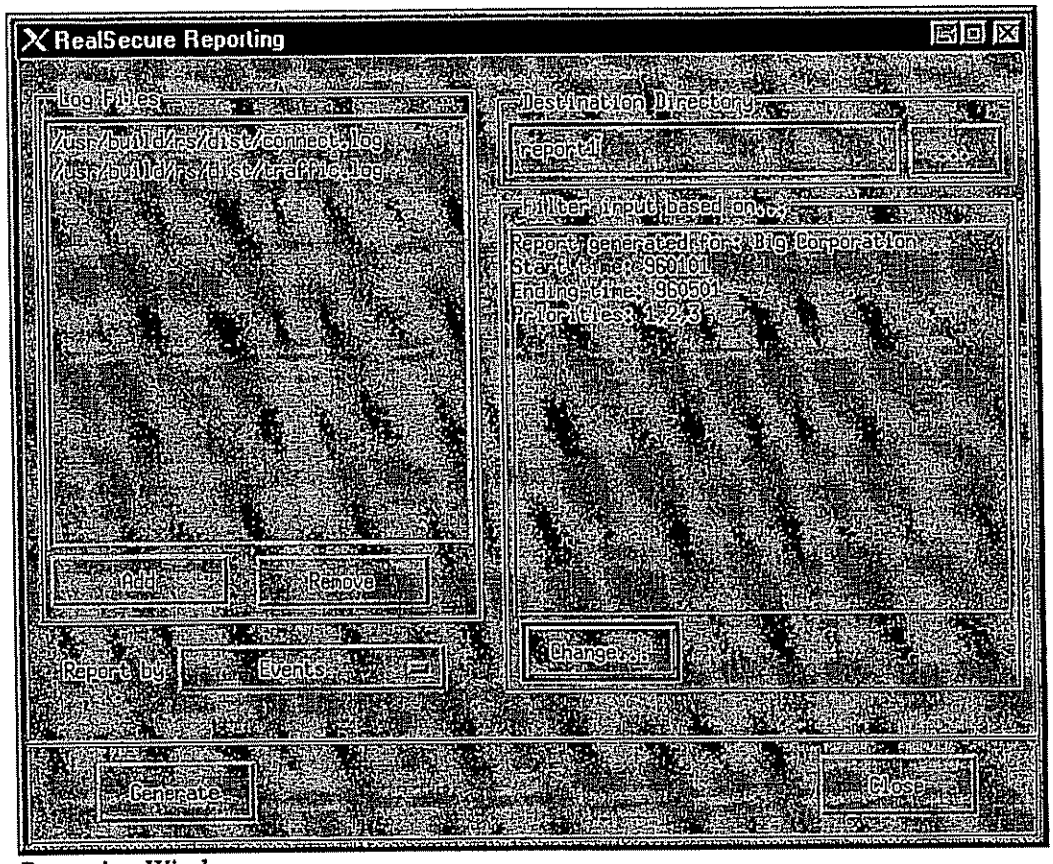


Playback Window

Select the one you wish to view and click the **Play** button. The play speed can be modified with the slider at the bottom of the screen. Pressing **Stop** gets you back to the list of connections.

Reporting Feature

The reporting feature uses logs generated with the **Log Info** action. To generate reports, click on the **Reports** button on the RealSecure toolbar. This displays the Reporting window.



Reporting Window

The Reporting window allows you to configure what log files are used to generate the report, where the report will be stored, and options to set for generating the report.

On the left of the screen is the list of log files to use to generate the report. To add another file to the list, click on the **Add** button and select the file you wish to use. To remove a file from the list, click on the name in the list and click on **Remove**. The **Report by** menu selects one of the three reports to generate: by source address, by destination address, and by event type (the default).

The destination directory for the reports can be changed by clicking on the **...** button. If you select a directory that doesn't exist, RealSecure will create it for you.

The other list shows what parts of the logs will be used in reporting. For instance, you can generate a

report on only certain priorities of events, certain time periods, and certain addresses. To modify this list, click on the **Change** button.

The **Change** button brings up a series of settings for report generation. The **Title** option allows you to specify a "Report Generated for..." title in the report. The **Source** and **Destination Addresses** fields allow you to specify an IP or range of IPs to limit the report. The **Start** and **End Times** fields allow you to specify a time period to report on, in yymmddhhmm.ss format (year, month, day, hour, minute, second). Here is one valid configuration:

```
Report generated for:  Internet Security Systems
Start date: 9610250900
End date: 9611250900
Source addresses: 10.0.0.1,20.0.0.1-20.0.0.120
Destination addresses: Ken,Barbie
```

This generates a report on all events from Oct 25, 1996 at 9 am to Nov 25, 1996 at 9 am. It only shows events coming from the host 10.0.0.1 and hosts 20.0.0.1 to 20.0.0.120 that are destined for the two hosts Ken and Barbie.

After setting up your report, click on **Generate** to generate a report. After a short delay, a browser will pop up on the report title page. If not, you will receive an error message explaining why.

Back to the Main Page

RealSecure Features and Attack Signatures

IP Fragmentation

An IP packet is sometimes split into several fragments when it is transmitted over the network. These fragments are then reassembled at the destination to form a full IP packet. Some routers that filter out packets based on information in the TCP header rely on the information in the first fragment and then blindly pass the remaining fragments. It is possible to construct individual fragments of an IP packet so that subsequent packets overlap and, as a result, overwrite parts of the TCP header when they are reassembled at the destination. The result of this is that an intermediate filtering router is tricked into

believing that a packet is destined for a service that is allowed, when in reality the packet is destined for a service that would normally be filtered out.

Ping Flooding

A Ping Flood is an attempt to saturate a network with packets in order to slow or stop legitimate traffic going through the network. A continuous series of ICMP Echo Requests are made to a target host on the network, which then responds with an ICMP Echo Reply. The continuing combination of requests and replies will slow the network and cause legitimate traffic to continue at a significantly reduced speed or, in extreme cases, to disconnect.

Arp Check

ARP, Address Resolution Protocol, is used to determine the Ethernet address of a machine on a network given its IP address. If an ARP is received for a machine on the network, it will then immediately send a reply. If the machine the ARP is destined for has crashed or otherwise disconnected from the network, several ARPs will be sent to it without any response. This lack of response to ARP packets is used to determine if a machine on the network has crashed.

IP Duplicate Check

Only one machine on a network should send packets with a specific IP address. If a second machine on the network starts to send packets claiming to have the same source address, a network problem has occurred. A machine on the network may be misconfigured to have the same IP address as another machine, causing network conflicts. The other possibility is that a machine on the network may be sending out IP packets with a forged source address.

IP Half Scan

A standard TCP connection is established by sending a SYN packet to the destination host. If the destination is listening for a connection on the specified port, it will respond with a SYN/ACK packet. The initial sender then replies to the SYN/ACK with an ACK packet, and the connection is established. If the destination host is not listening for a connection on the specified port, it will respond with an RST packet instead of a SYN/ACK. Most system logs do not log that a connection is made until the final ACK packet is received from the source. Sending an RST packet instead of the final ACK results in the connection never actually being established; so no logging takes place. But, because the source can identify whether the destination host sent a SYN/ACK or an RST, an attacker can determine exactly what ports are open for connections, without the destination ever being aware that it has been probed.

IP Unknown Protocol

A standard IP packet has an 8-bit protocol field in it. Common values for this field include 6 (TCP), 17 (UDP), and 1 (ICMP). Attackers sometimes use a non-standard value for this field in order to exchange data between machines without logging mechanisms detecting the data that is being transmitted.

UDP Bomb

A UDP packet that is constructed with illegal values in certain fields will cause some older operating systems to crash when the packet is received. If the target machine does crash, it is often difficult to

determine the cause of the crash. Most operating systems that are not vulnerable to this problem will silently discard the invalid packet, leaving no traces that it was being subjected to a malicious attack.

SYN Flood

A standard TCP connection is established by sending a SYN packet to the destination host. If the destination is listening for a connection on the specified port, it will respond with a SYN/ACK packet. The initial sender then replies to the SYN/ACK with an ACK packet, and the connection is established. When the SYN/ACK is sent back to the source, a block of memory is allocated to hold information about the state of the connection that is currently being established. Until the final ACK is received or a timeout is reached, this block of memory sits unused waiting for more information to be received from the source host. By sending numerous SYN packets to a host, the destination will exhaust the portion of memory it has set aside to deal with opening connections, and legitimate connections will no longer be able to connect to the host. This situation can be detected by the flood of SYN packets without accompanying responses. It can be corrected by sending the destination RST packets that correspond to the initial SYNs. This results in the destination host freeing up that block of memory and making room for a new legitimate connection.

Source Routing

IP packets sent over the Internet are normally sent between different routers in order to reach their final destination. The route each packet takes is determined dynamically by each router along the way. Enabling the source routing option on an IP packet allows the packet itself to tell each router the path it wishes to take to reach its final destination. By routing packets through a path that bypasses filtering routers and other normal security mechanisms, an attacker may be able to reach a host that would normally not be reached. Also, it can be used to authenticate an intruder to systems that rely on the source IP address for access control.

Rlogin Decoding

An Rlogin connection allows a user to remotely login to a host without a password by using a trust relationship between the account on the source machine and the account on the destination host. The source machine and username, along with the destination machine and username are logged with this feature.

Rlogin -froot Vulnerability Check

If a remote user passes the name `-froot` to `rlogin` to a machine, certain operating systems will bypass normal security mechanisms and log the user in directly as `root`. This vulnerability allows anyone who can access the `rlogin` service on the target host to gain immediate `root` access to the machine.

FTP Username Decoding

FTP, File Transfer Protocol, allows users to transfer files between machines. Username decoding discovers the name of the account that is being used to transfer files across the network.

FTP Password Decoding

FTP passes a plain text password across the network in order to authenticate that a user has access to the

files on the destination host. This password is discovered using FTP password decoding. This allows an administrator to log invalid password attempts, check passwords for strength against attack, and keep complete logs of activity.

FTP Site Command Decoding

The FTP site command allows a user to execute certain commands on a destination host in addition to the normal FTP facility of transferring files. In ordinary usage of FTP, this is not a commonly used command. While there may be a legitimate reason to execute site commands under certain circumstances, this facility has also been used to gain access. Consequently, an administrator may wish to view and log the site commands being executed to check for possible abuse.

FTP GET File Decoding

Files being transferred from the destination host to the source host use a GET command in order to transfer the files. FTP GET decoding discovers all files that are being transferred to the source host over FTP.

FTP PUT File Decoding

Files being transferred from the source host to the destination host use a PUT command in order to transfer the files. FTP PUT decoding discovers all files that are being transferred to the destination host over FTP.

FTP Mkdir Decoding

FTP allows a user to create a new directory on the target machine. FTP Mkdir decoding discovers all new directories that are created through FTP.

FTP CWD ~root Vulnerability Check

Certain versions of the FTP daemon allow access to files on a machine through a sequence of commands culminating with `CWD ~root`. This vulnerability allows an attacker who can access FTP on the target host to transfer files that he/she would not normally have access to.

HTTP GET Decoding

Pages, images, and all other information that is viewed through a Web browser on the World Wide Web are transferred through HTTP using the GET command. HTTP GET decoding discovers all Web pages that are being transmitted unsecurely to a machine. This allows an administrator to track, log, and view Web traffic on the network.

HTTP PHF Vulnerability Check

The `cgi-bin` script `PHF`, which comes preinstalled with several versions of NCSA and Apache Web servers, contains a vulnerability that allows anyone who can access your Web site to gain access to your machine.

Ident User Decoding

The Ident port is used by services to identify the account by which a connection is being made on a machine. This can be used to track a connection back to a specific user on a multi-user machine.

Ident Buffer Overflow Vulnerability Check

Certain programs that connect back to the `ident` service to log user information expect a properly formatted response. If the response is longer than expected, the buffer the response is read into is overflowed, allowing the remote user to execute commands on the host machine.

Ident Newline Vulnerability Check

Certain programs that connect back to the `ident` service to log user information expect a properly formatted response. If the response contains newlines, the response may be improperly parsed, allowing the remote user to execute commands on the host machine.

POP Username Decoding

The POP service is used by numerous e-mail programs to retrieve e-mail from a mail server and read it on a local machine. POP username decoding discovers the username of the user who is reading mail through the POP service.

POP Password Decoding

POP password decoding discovers all successful and unsuccessful passwords that a user attempts to use to login to a mail server using POP.

RSH Decoding

RSH, the remote shell command, allows a user to execute a shell command over the network using a trust relationship between the user on the local machine and the user account on the remote machine. RSH decoding discovers both the local and remote usernames as well as the command that is being executed.

E-Mail From

This decoding discovers the sender of all mail that is sent over the network using SMTP.

E-Mail To

This decoding discovers the recipient of all mail that is sent over the network using SMTP.

E-Mail Subject

This decoding discovers the subject line of all mail that is sent over the network using SMTP.

E-Mail VRFY

The VRFY command is used to verify if a user on a remote system exists. This is sometimes used

legitimately to determine if the recipient of a message at the intended destination will be able to receive the message. It is also sometimes used to gain information about users on a system by trying to find out if certain common account names exist on a machine.

E-Mail EXPN

The EXPN command is used to expand the address of a user on a remote system. This is sometimes used legitimately to determine the full address of an intended mail recipient. It is also sometimes used to gain information about users on a system by trying to find out if certain common account names exist on a machine.

E-Mail WIZ Vulnerability Check

The WIZ command in Sendmail existed to allow access to a machine under certain circumstances. It is no longer present in current versions of Sendmail, but old versions still in use may allow an attacker to gain `root` access to a machine by using this command.

E-Mail DEBUG Vulnerability Check

The DEBUG command in Sendmail existed to allow debugging of a remote Sendmail daemon. It is no longer present in current versions of Sendmail, but old versions still in use allow an attacker to gain `root` access to a machine by using this command remotely.

E-Mail Pipe Vulnerability Check

By inserting a pipe (|) character into certain fields in an e-mail, Sendmail may be forced to execute a command on the remote host. This results in a remote attacker being able to execute commands as `root` on the machine.

E-Mail Decode Vulnerability Check

By sending mail to decode or uudecode alias that is present in some systems, a remote attacker may be able to create or overwrite files on the remote host.

Back to Main Page

Project Managers: Keith Cooley and Christopher Klaus

Marketing Manager: Patrick Taylor

Original Design and Code: Nate Lawson

Lead Developer: David J. Meltzer

Back to Main Page